



AUDIT OF WALLET ANDROID APPLICATION



August 18th 2023 | v. 1.0

TECHNICAL SUMMARY

MarsDao engaged Zokyo to conduct a security assessment on their Android Mobile application beginning on June 5th and ending on August 18th, 2023. MarsDao (MDAO) Wallet is the most user-friendly cryptocurrency wallet. Send, receive, and store Bitcoin and many other cryptocurrencies and digital assets safely and securely with the MDAO Wallet mobile app. The security assessment was scoped to the Android mobile application (com.ttmbank.wallet.app). An audit of the security risk and implications regarding the changes introduced by the development team at MarsDAO prior to its production release, shortly following the assessment deadline. Though this security audit's outcome is satisfactory, only the most essential aspects were tested and verified to achieve objectives and deliverables set in the scope due to time and resource constraints. It is essential to note the use of the best practices for secure Mobile application development.



Table of Contents

Auditing Strategy and Techniques Applied	3
Executive Summary	5
Structure and Organization of the Document	6
Complete Analysis	7

AUDITING STRATEGY AND TECHNIQUES APPLIED

Zokyo performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy in regard to the scope of the penetration test. The majority of the time was spent evaluating its use of mnemonic seed Protection.

The following phases and associated tools were used throughout the term of audit:

- Research into architecture, purpose, and use of client wallet.
- Manual code read and analysis.
- Reverse engineering of the hashing and encryption functions used Inside the wallet.
- Scanning of code used to locate bugs or security flaws.(MOBSF)
- Proxying the traffic from the local client to the external Internet to determine the traffic and data leaving the system. (REDUX, POSTMAN, BURP SUITE)

The audit is focused on various aspects to ensure the security of the mobile application and includes:

- Implementation of correctness and adherence to industry best practices.
- Exposure of critical information during user interactions, including authentication mechanisms.
- Adversarial actions and attacks that could impact funds, such as draining or manipulating funds.
- Proper management of funds via transactions to prevent mismanagement.
- Identification and remediation of vulnerabilities in the code, as well as ensuring secure interaction between the related and network components.
- Secure management of encryption and storage of private keys, including the key derivation process.
- Prevention of inappropriate permissions and excess authority.
- Ensuring data privacy, prevention of data leakage, and maintaining information integrity.
- Identification and remediation of any other potential security risks, as identified during the initial analysis phase.

In summary, Zokyo identified a few security risks and recommends performing further testing to validate extended safety and correctness in context to the whole structure

SCOPE :

The following scope was audited by Zokyo team :

Android Application : <https://play.google.com/store/apps/details?id=com.ttmbank.wallet.app&hl=us&gl=US&pli=1>

Application Name: com.ttmbank.wallet.app (MDAO)

Version : 2.2.4



Executive Summary

There were three medium issues found during the audit and some low severity and one information issue.

They are described in detail in the “Complete Analysis” section.

STRUCTURE AND ORGANIZATION OF THE DOCUMENT

For the ease of navigation, the following sections are arranged from the most to the least critical ones. Issues are tagged as “Resolved” or “Unresolved” or “Acknowledged” depending on whether they have been fixed or addressed. Acknowledged means that the issue was sent to the MarsDAO team and the MarsDAO team is aware of it, but they have chosen to not solved it. The issues that are tagged as “Verified” contain unclear or suspicious functionality that either needs explanation from the Client or remains disregarded by the Client. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

 **Critical**

The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

 **High**

The issue affects the ability of the contract to compile or operate in a significant way.

 **Medium**

The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.

 **Low**

The issue has minimal impact on the contract's ability to operate.

 **Informational**

The issue has no impact on the contract's ability to operate.

COMPLETE ANALYSIS

FINDINGS SUMMARY

#	Title	Risk	Status
1	Root check and Safety-net check not implemented	Medium	Resolved
2	SSL Pinning, Proxy and VPN check not implemented	Medium	Resolved
3	Insufficient Anti Reverse Engineering Detection	Medium	Resolved
4	Seed phrase is displayed without a timeout	Low	Resolved
5	Lack of token Name, Symbol Validation	Low	Unresolved
6	Lack ScreenShot/ScreenRecording Detection	Low	Resolved
7	Background obfuscation issue	Low	Resolved
8	Lack of Clipboard Security Reminders during Wallet Export	Informational	Acknowledged

Root check and Safety-net check not implemented

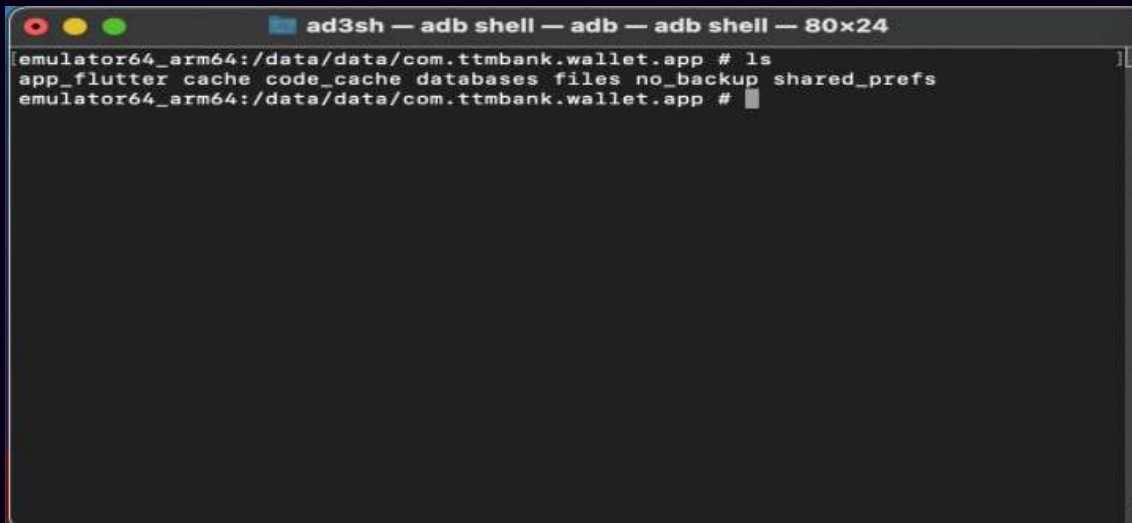
Description :

The MDAO app is susceptible to exploitation due to the absence of a proper root check and Safety-Net check during installation. As a result, if the device is rooted and SuperSU is installed, the "/data/data/com.ttmbank.wallet.app" files become accessible, compromising the security of the app's data information and code.

Impact:

An attacker with root access to the device can manipulate and access sensitive data stored by the Mado app. This includes potentially confidential user information, application code, and any other data associated with the app's functionalities. This vulnerability poses a significant risk to the confidentiality and integrity of user data.

Proof of concept:

A screenshot of an Android Studio interface showing an adb shell session. The title bar reads "ad3sh — adb shell — adb — adb shell — 80x24". The terminal window shows the command prompt "emulator64_arm64:/data/data/com.ttmbank.wallet.app #". The user has entered the command "ls", and the output is "app_flutter cache code_cache databases files no_backup shared_prefs". The prompt is now "emulator64_arm64:/data/data/com.ttmbank.wallet.app #".

```
ad3sh — adb shell — adb — adb shell — 80x24
emulator64_arm64:/data/data/com.ttmbank.wallet.app # ls
app_flutter cache code_cache databases files no_backup shared_prefs
emulator64_arm64:/data/data/com.ttmbank.wallet.app #
```

Recommendation:

Implement a thorough root check during the installation process to detect if the device has been rooted. This can be achieved by using root-detection libraries or implementing custom checks.

Root check and safety-net check must be exist

Comment: Client added Warning message if the MarsDao Installed in rooted Device

SSL Pinning, Proxy and VPN check not implemented

Description :

During analysis of the MDAO Android application, it has been observed that SSL pinning is not enabled. As a result, it is possible to intercept all requests and responses made by the app. This poses a significant security risk as an attacker can manipulate or access sensitive data transmitted between the app and the backend server.

Proof of Concept

Step 1:

Configure mobile proxy through burp-suite Proxy tool.

Step 2:

Install MDAO app below build in rooted device.

```
$ adb install MDAO.apk
```

Step 3:

Navigate app functionality and observe app traffic in burp-suite proxy tool.

Impact:

App traffic can be manipulate/access in burp-suite proxy tab.

Attacker can perform Api's related attacks

Recommendation:

SSL/TLS pinning must be implemented in a proper way.

Proxy check must be implemented

Insufficient Anti Reverse Engineering Detection

Description :

During the analysis of the MDAO app, it has been identified that the app lacks proper detection mechanisms for widely used reverse engineering tools. Specifically, the app fails to detect the presence of popular tools such as Magisk Manager and Xposed Framework. This vulnerability exposes the app to potential misuse and unauthorized access by individuals attempting to analyze or manipulate the app's code and functionality.

Proof of Concept:

1. Install Magisk Manager and Xposed Installer app on the testing device.
2. Launch the MDAO app.

Result:

The MDAO app does not terminate or provide any warning or notification to the user regarding the presence of reverse engineering tools such as Magisk Manager and Xposed Framework. This allows attackers or malicious users to potentially exploit the app's vulnerabilities without detection.

Impact:

The lack of proper detection mechanisms for reverse engineering tools poses several risks to the security and integrity of the MDAO app:

1. Unauthorized Access: Attackers can exploit the app's vulnerabilities to gain unauthorized access to sensitive information, manipulate data, or perform unauthorized actions.
2. Code Manipulation: Reverse engineering tools can be used to analyze and modify the app's code, which may lead to the introduction of malicious code or unauthorized modifications to the app's functionality.

3. Data Leakage: Attackers can extract confidential user data, intellectual property, or other sensitive information from the app, potentially leading to privacy breaches or reputational damage.

Recommendation:

To mitigate the risks associated with inadequate detection of reverse engineering tools, the following recommendations should be implemented in the MDAO app:

1. Detection Mechanisms: Develop robust detection mechanisms within the app to identify the presence of widely used reverse engineering tools. This can be achieved by monitoring associated application packages, files, processes, or other tool-specific modifications and artifacts.
2. Specific Tool Checks: Implement specific checks for popular reverse engineering tools such as Xposed Framework, Frida Server, and Magisk Manager. The app should respond appropriately when the presence of any of these tools is detected, such as providing warnings or notifications to the user and potentially terminating the app.
3. Regular Updates: Stay up-to-date with the evolving landscape of reverse engineering tools and techniques, and update the detection mechanisms accordingly. Regularly assess new tools and modify the detection mechanisms to maintain their effectiveness

Seed phrase is displayed without a timeout

Description:

Wallets are able to show the seed phrase to the user, and since there's no timeout for this view, once the phrase is shown, it stays uncovered indefinitely. A careful user could partially mitigate this issue by setting a global lockout timeout for the phone; however, the wallet should implement any available means to secure the mnemonic.

Exploit Scenario:

A wallet user leaves the phone unattended with the seed phrase view on. An attacker takes the victim's phone, reads the recovery phrase, and uses it later to take over the victim's wallet.

Recommendation:

Add a timeout to the seed phrase view that exits the view or hides the secret

Lack of token Name, Symbol Validation

Description:

Mdao wallet allows users to import custom tokens, and display the token name and symbol in the wallet UI. The ERC20 token contract standard doesn't have any restrictions on any of the token properties, and anyone can deploy token contracts on the blockchain. Once a token is added, the wallet will fetch the information and display them in its interface.

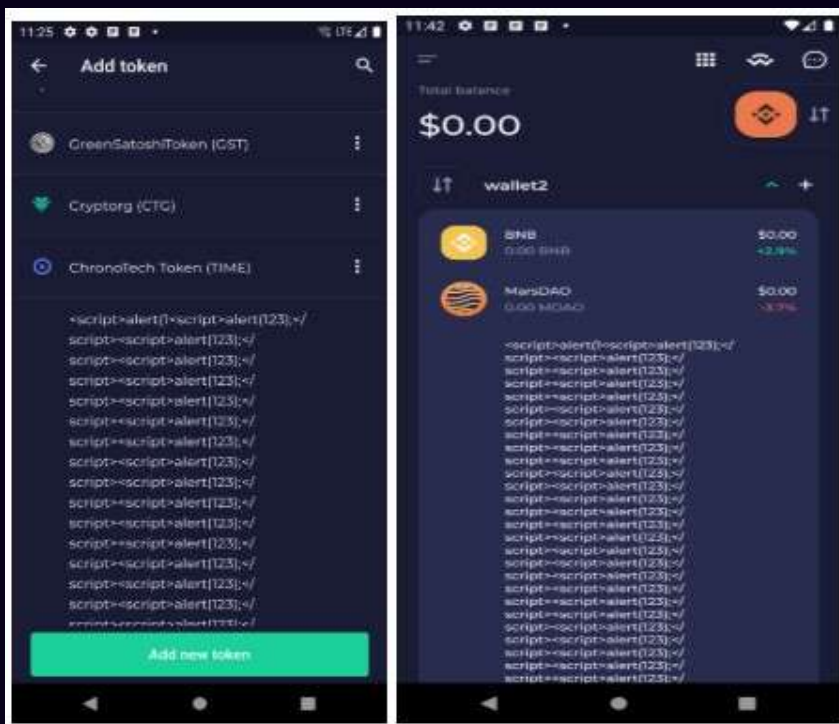
Impact:

Importing a crafted malicious token can bring a negative user experience to the user, and the attacker can potentially craft a message to perform a phishing attack

Reproduce Steps

Import the following token "0xe9bb17e159Ff8551e08616Fd310192ea57BBE52b" in the BSC Smart chain

Proof of Concept:



Recommendation:

It is recommended to place restriction on token name and symbol in add new token functionality

Lack ScreenShot/ScreenRecording Detection

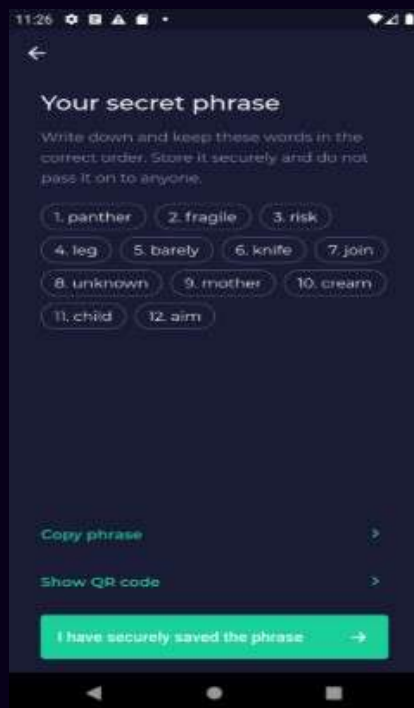
Description:

The current version of the wallet application does not implement screenshot/screen recording detection. This absence creates a potential vulnerability that could be exploited by malicious applications.

Impact:

A malicious application accessing an image of the mnemonic phrase could result in loss of all wallet funds. In addition, account balance or transaction history leaks are a breach of user privacy.

Proof of concept:



Recommendation:

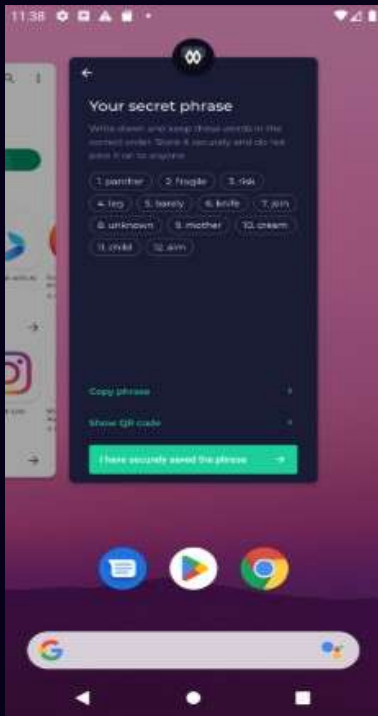
It is recommended to add screenshot/screen recording detection and prohibit screenshot/screen recording

On Android, we recommend setting FLAG_SECURE on the application window and making sure that no private content is shown in other windows. For more information, we suggest referring to this blog post about vulnerabilities of weak screenshot protection. <https://blog.doyensec.com/2019/08/22/modern-password-managers-flag-secure.html>

Background obfuscation issue

Description:

App UI is not obfuscation when the app is in the background. If the wallet is being exported, the mnemonic phrase may be leaked.



Recommendation:

It is recommended to add an obfuscation mechanism to avoid sensitive data leakage.

Lack of Clipboard Security Reminders during Wallet Export

Description :

When exporting wallets, users are allowed to copy mnemonic phrases and the app lacks security reminders, which may be subject to clipboard hijacking attacks.

Recommendation:

It is recommended to remind users that they should record by transcribing instead of directly using the clipboard for copying

We are grateful for the opportunity to work with the MarsDAO team.

The statements made in this document should not be interpreted as an investment or legal advice, nor should its authors be held accountable for the decisions made based on them.

Zokyo Security recommends the MarsDAO team put in place a bug bounty program to encourage further analysis of the wallet application by third parties.

